

REMARKS

In response to the Office Action dated December 13, 2007, Applicants respectfully request reconsideration based on the above claim amendments and the following remarks. Applicants respectfully submit that the claims as presented are in condition for allowance. Prior to entry of this response, Claims 1-6, 8-12, 14-16, 18, and 20 were pending in the application, of which Claims 1, 8, and 14 are independent. In the Office Action dated December 13, 2007, Claims 1-6, 8-12, 14-16, 18, and 20 were rejected under 35 U.S.C. § 112 and § 103(a). Following this response, Claims 1-3, 8-9, 14-16, 18, and 20 remain in this application with Claims 4-6 and 10-12 being canceled without prejudice or disclaimer. Applicants hereby address the Examiner's rejections in turn.

I. **Rejection of the Claims Under 35 U.S.C. § 112, Second Paragraph**

In the Office Action dated December 13, 2007, the Examiner rejected Claims 1-18 and 20 under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which the Applicants regard as their invention. Specifically, the Examiner stated that the limitation "a form name corresponding to a version number corresponding to each of the one or more field names of the form" was unclear. Applicants have amended Independent Claims 1, 8, and 14 to recite "one or more field names, the form name corresponding to each of the one or more field names of the form, the data type corresponding to each of the one or more field names of the form, and the version number corresponding to each of the one or more field names of the form." Applicants respectfully submit that the amendments to Claim 1, 8, and 14 overcome this rejection and add no new matter.

II. Rejection of Claims 1-18 Under 35 U.S.C. § 103(a)

In the Office Action, the Examiner rejected Claims 1-3, 8-9, 14-16, and 18 as being unpatentable over U.S. Patent No. 5,999,948 ("*Nelson*") in view of U.S. Patent No. 6,496,843 ("*Getchius*"). Claims 1, 8, and 14 have been amended, and Applicants respectfully submit that the amendments overcome this rejection and add no new matter.

Amended Claim 1 is patentably distinguishable over the cited art for at least the reason that it recites, for example, "the data identifying the one or more fields of the form comprising a name, a data type, and a version number for each of the one or more fields of the form" and "each of the one or more field names are associated with a corresponding data type indicating a type of input field to display for each of the one or more field names." Amended Claims 8 and 14 each includes a similar recitation. Support for these amendments can be found in the specification at least on page 10, lines 7-18.

Consistent with embodiments of the invention, a field engine table may be utilized by a database access class to generate a markup language for displaying a form. (See specification, page 10, lines 4-7.) The database access class may be called from a JSP page. (See specification, page 10, line 7.) In particular, the field engine table may define field names for each of the fields in a form and a data type for each field. (See specification, page 10, lines 8-9.) To accomplish this, the field engine table may contain a form name field, a field name field, a version number field, and a form type field. (See specification, page 10, lines 9-11.) The data type field may comprise

data indicating what type of input field should be displayed. (See specification, page 10, lines 14-15.)

In contrast, *Nelson* at least does not disclose the aforementioned recitation from Claim 1. *Nelson* merely discloses registering form description language files with a dynamic forms software module. (See *Nelson*, Abstract.) For example, *Nelson* discloses factory objects that do nothing more than create instances of specific object types, such as Windows Combo Box widgets. (See *Nelson*, col. 7, lines 25-27.) *Nelson* states that there are only a few types of data and a limited number of ways to present each data piece and that form behavior is completely determined by the form description. (See *Nelson*, col. 7, lines 31-39.) As stated by the Examiner, *Nelson* does not disclose utilizing a field engine table to retrieve one or more field names of a form. (See Office Action, page 4, lines 3-4.) Applicants respectfully submit that *Nelson* also does not disclose utilizing the field engine table to retrieve a data type corresponding to field names of the form to indicate the type of input field to be displayed. Rather, *Nelson* merely discloses creating a widget with a generic factory object by selecting from a limited number of data presentation methods.

Furthermore, *Getchius* does not overcome *Nelson*'s deficiencies. *Getchius* merely discloses that an external process may copy blob data from multiple tables in which an associated field name differ with each table. (See col. 52, lines 31-33.) The external process uses the data included in a temporary table to access the blob data associated with a particular table name and field name to subsequently index into each particular table name using the identifier to extract the actual blob data. (See col. 52, lines 38-42.) Therefore, *Getchius*, fetches blob data associated with a particular table

and field name. However, nowhere in *Getchius* does it disclose utilizing a field engine table to retrieve data types corresponding to field names to indicate the type of input field to be displayed. Rather *Getchius* fetches blob data associated merely with a particular table name and field name.

Combining *Nelson* with *Getchius* would not have led to the claimed invention because *Nelson* and *Getchius*, either individually or in combination, at least do not disclose "the data identifying the one or more fields of the form comprising a name, a data type, and a version number for each of the one or more fields of the form" and "each of the one or more field names are associated with a corresponding data type indicating a type of input field to display for each of the one or more field names," as recited by amended Claim 1. Amended Claims 8 and 14 each includes a similar recitation. Accordingly, independent Claims 1, 8, and 14 each patentably distinguishes the present invention over the cited references, and Applicants respectfully request withdrawal of this rejection of Claims 1, 8, and 14.

Dependent Claims 2-3, 9, 15-16, and 18 are also allowable at least for the reasons described above regarding independent Claims 1, 8, and 14, and by virtue of their respective dependencies upon independent Claims 1, 8, and 14. Accordingly, Applicants respectfully request withdrawal of this rejection of dependent Claims 2-3, 9, 15-16, and 18.

III. Rejection of the Claim 20 Under 35 U.S.C. § 103(a)

In the Office Action, the Examiner rejected Claim 20 as being unpatentable over *Nelson* in view of *Getchius* and further in view of U.S. Patent No. 6,718,515 ("*Conner*").

Claim 20 is patentably distinguishable over the cited art for at least the reason that it includes, due to its dependency on amended independent Claim 1, for example, "the data identifying the one or more fields of the form comprising a name, a data type, and a version number for each of the one or more fields of the form" and "each of the one or more field names are associated with a corresponding data type indicating a type of input field to display for each of the one or more field names." Support for this amendment can be found in the specification at least on page 10, lines 7-18.

Consistent with embodiments of the invention, and as stated above, a field engine table may be utilized by a database access class to generate a markup language for displaying a form. (See specification, page 10, lines 4-7.) The database access class may be called from a JSP page. (See specification, page 10, line 7.) In particular, the field engine table may define field names for each of the fields in a form and a data type for each field. (See specification, page 10, lines 8-9.) To accomplish this, field engine table may contain a form name field, a field name field, a version number field, and a form type field. (See specification, page 10, lines 9-11.) The data type field comprises data indicating what type of input field should be displayed. (See specification, page 10, lines 14-15.)

In contrast, and as stated above, *Nelson* at least does not disclose the aforementioned recitation. *Nelson* merely discloses registering form description language files with a dynamic forms software module. (See *Nelson*, Abstract.) *Nelson* discloses factory objects that do nothing more than create instances of specific object types, such as Windows Combo Box widgets. (See *Nelson*, col. 7, lines 25-27.) *Nelson* states that there are only a few types of data and a limited number of ways to present

each data piece and that form behavior is completely determined by the form description. (See *Nelson*, col. 7, lines 31-39.) As stated by the Examiner, *Nelson* does not disclose utilizing a field engine table to retrieve one or more field names of a form. (See Office Action, page 4, lines 3-4.) Applicants respectfully submit that *Nelson* also does not disclose utilizing the field engine table to retrieve a data type corresponding to field names of the form to indicate the type of input field to be displayed. Rather, *Nelson* merely discloses creating a widget with a generic factory object by selecting from a limited number of data presentation methods.

As further stated above, *Getchius* does not overcome *Nelson's* deficiencies. *Getchius* merely discloses that an external process may copy blob data from multiple tables in which an associated field name differ with each table. (See col. 52, lines 31-33.) The external process uses the data included in a temporary table to access the blob data associated with a particular table name and field name to subsequently index into each particular table name using the identifier to extract the actual blob data. (See col. 52, lines 38-42.) Therefore, *Getchius*, fetches blob data associated with a particular table and field name. However, nowhere in *Getchius* does it disclose utilizing a field engine table to retrieve data types corresponding to field names to indicate the type of input field to be displayed. Rather *Getchius* fetches blob data associated merely with a particular table name and field name.

Also, *Conner* does not overcome *Nelson's* and *Getchius's* deficiencies. *Conner* merely discloses creating a table format object and using the object to generate an HTML table as a dynamic page in response to a client browser. (See col. 5, lines 11-14.) A routine in *Conner* begins by creating a table format object called a

tableFormatter. (See col. 5, lines 14-16.) The object is created during a page authoring process. (See col. 5, lines 16-17.) In response to a client request, the request object and data object are passed, in *Conner*, to the tableFormatter that formats the table for use in a page. (See col. 5, lines 38-44.) In other words, in response to a client request in *Conner*, a .jsp servlet creates the HTML table. Then the servlet populates the table according to properties set in the tableFormatter that is hard-coded by a page author. *Conner* populates a table according to properties that are hard-coded and not retrieved. *Conner*, therefore, cannot disclose utilizing a field engine table to retrieve data types corresponding to field name to indicate the type of input field to be displayed. Rather, *Conner* merely discloses creating a table in a web page from hard-coded properties.

Combining *Nelson* with *Getchius* and *Conner* would not have led to the claimed invention because *Nelson*, *Getchius*, and *Conner*, either individually or in combination, at least do not disclose "the data identifying the one or more fields of the form comprising a name, a data type, and a version number for each of the one or more fields of the form" and "each of the one or more field names are associated with a corresponding data type indicating a type of input field to display for each of the one or more field names," as included in dependent Claim 20 by virtue of its dependency on amended independent Claim 1. Accordingly, dependent Claim 20 patentably distinguishes the present invention over the cited references, and Applicants respectfully request withdrawal of this rejection of Claim 20.

IV. Conclusion

In view of the foregoing remarks, Applicants respectfully request the reconsideration and reexamination of this application and the timely allowance of the

pending claims. The preceding arguments are based only on the arguments in the Office Action, and therefore do not address patentable aspects of the invention that were not addressed by the Examiner in the Office Action. The claims may include other elements that are not shown, taught, or suggested by the cited art. Accordingly, the preceding argument in favor of patentability is advanced without prejudice to other bases of patentability. Furthermore, the Office Action contains a number of statements reflecting characterizations of the related art and the claims. Regardless of whether any such statement is identified herein, Applicants decline to automatically subscribe to any statement or characterization in the Office Action.

Please grant any extensions of time required to enter this response and charge any additional required fees to our deposit account 13-2725.

Respectfully submitted,
MERCHANT & GOULD P.C.

P.O. Box 2903
Minneapolis, MN 55402-0903
404.954.5066

Date: March 13, 2008

/D. Kent Stier/
D. Kent Stier
Reg. No. 50,640

DKS:sm

39262

PATENT TRADEMARK OFFICE